

Bloch_law_plot_V2

oscar

2025-06-12

Import the Data and organize them

First we import the data of each tasks for one participant and each r library that we need.

```
library(readr)
library(ggplot2)
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'
## Les objets suivants sont masqués depuis 'package:stats':
##
##   filter, lag
## Les objets suivants sont masqués depuis 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggrepel)
```

#Change the number xx_faces_discr/BL_faces_xx_data_table.csv" and re-run all chunk to see the plot of

```
discr_landscape <- read_csv("/home/ruiz/Documents/Stage_brxl/DATA_BOCH_LAW/DIS/subj_03_bloch_law_landscape.csv")
```

```
## Rows: 500 Columns: 14
## -- Column specification -----
## Delimiter: ","
## chr  (2): subj, stimulus_filename
## dbl (12): trial_n, current_staircase, duration, contrast, stimulus_type, RT,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
discr_faces <- read_csv("/home/ruiz/Documents/Stage_brxl/DATA_BOCH_LAW/DIS/subj_03_bloch_law_faces_discr.csv")
```

```
## Rows: 500 Columns: 14
## -- Column specification -----
## Delimiter: ","
## chr  (2): subj, stimulus_filename
## dbl (12): trial_n, current_staircase, duration, contrast, stimulus_type, RT,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```

discr_objects <- read_csv("/home/ruiz/Documents/Stage_brxl/DATA_BOCH_LAW/DIS/subj_03_bloch_law_objects_03.csv")

## Rows: 500 Columns: 14
## -- Column specification -----
## Delimiter: ","
## chr (2): subj, stimulus_filename
## dbl (12): trial_n, current_staircase, duration, contrast, stimulus_type, RT,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
# we use head to check if each data set are imported properly.
head(discr_objects)

## # A tibble: 6 x 14
##   subj trial_n current_staircase duration contrast stimulus_type
##   <chr>   <dbl>           <dbl>    <dbl>    <dbl>         <dbl>
## 1 03         1             9    55505     56           0
## 2 03         2             9    55505     44           1
## 3 03         3             4    2924     56           1
## 4 03         4             1     500     56           0
## 5 03         5             3    1623     56           0
## 6 03         6             5    5268     56           0
## # i 8 more variables: stimulus_filename <chr>, RT <dbl>, resp <dbl>,
## #   correct <dbl>, updated_threshold <dbl>, updated_slope <dbl>,
## #   updated_lapse <dbl>, updated_guess <dbl>
head(discr_faces)

## # A tibble: 6 x 14
##   subj trial_n current_staircase duration contrast stimulus_type
##   <chr>   <dbl>           <dbl>    <dbl>    <dbl>         <dbl>
## 1 03         1             5    5268     56           0
## 2 03         2             5    5268     78           0
## 3 03         3             9    55505     56           1
## 4 03         4             6    9491     56           0
## 5 03         5             9    55505     78           1
## 6 03         6             3    1623     56           0
## # i 8 more variables: stimulus_filename <chr>, RT <dbl>, resp <dbl>,
## #   correct <dbl>, updated_threshold <dbl>, updated_slope <dbl>,
## #   updated_lapse <dbl>, updated_guess <dbl>
head(discr_landscape)

## # A tibble: 6 x 14
##   subj trial_n current_staircase duration contrast stimulus_type
##   <chr>   <dbl>           <dbl>    <dbl>    <dbl>         <dbl>
## 1 03         1             1     500     56           1
## 2 03         2             4    2924     56           0
## 3 03         3             1     500     78           1
## 4 03         4             6    9491     56           1
## 5 03         5             7   17100     56           1
## 6 03         6             7   17100     44           0
## # i 8 more variables: stimulus_filename <chr>, RT <dbl>, resp <dbl>,
## #   correct <dbl>, updated_threshold <dbl>, updated_slope <dbl>,
## #   updated_lapse <dbl>, updated_guess <dbl>

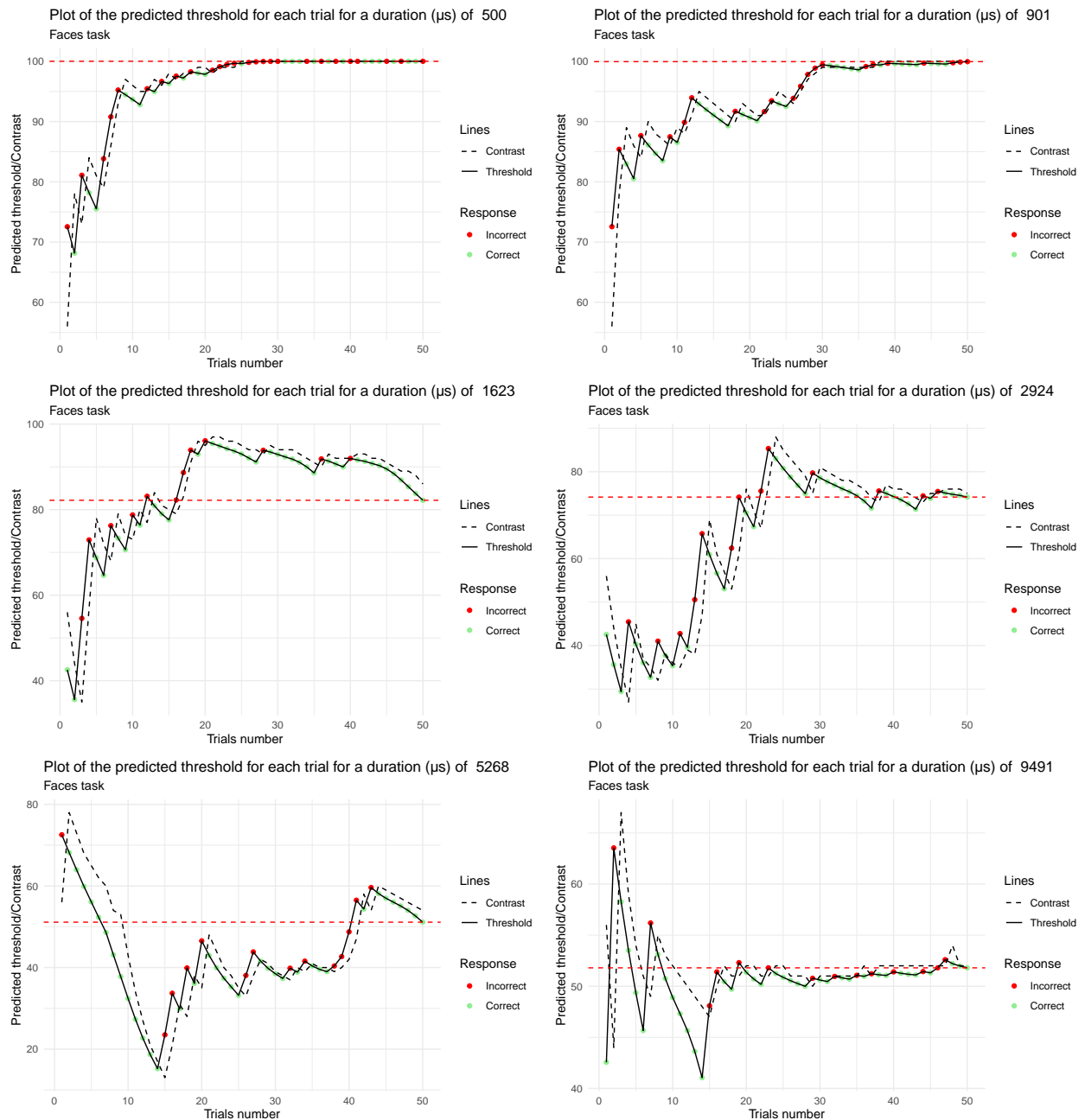
```

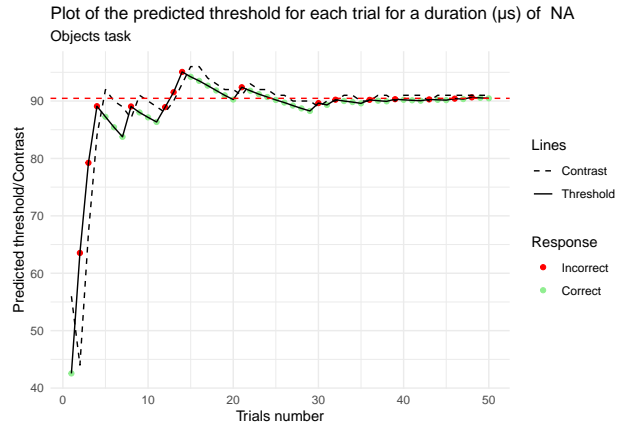
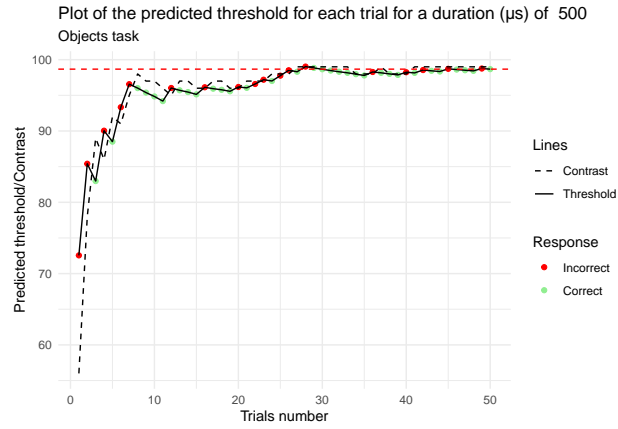
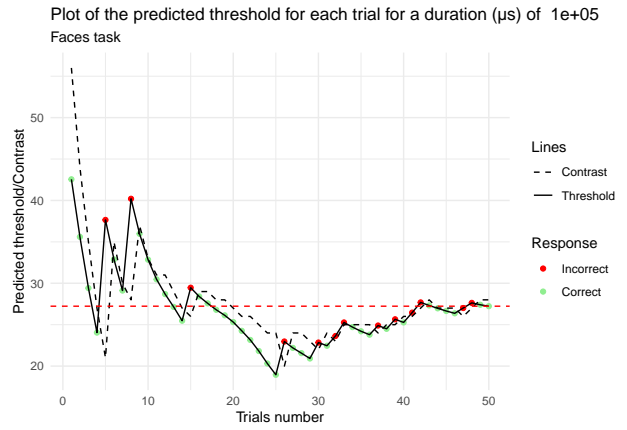
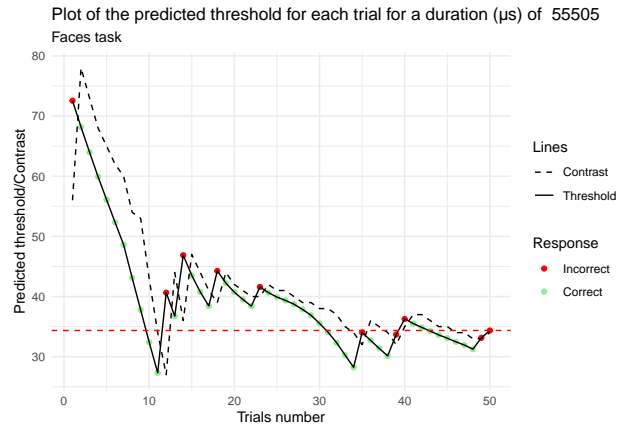
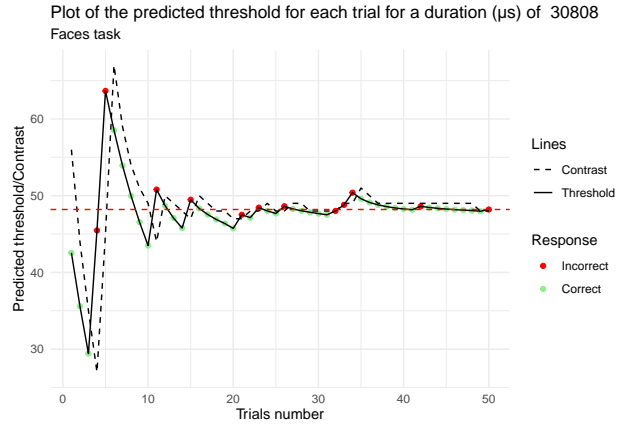
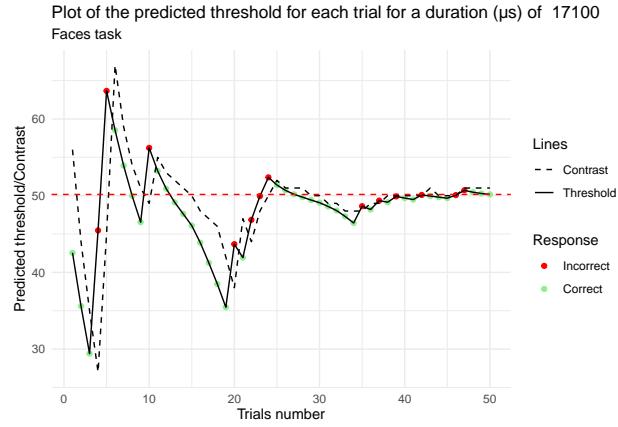
Dividing the data frames into subsets

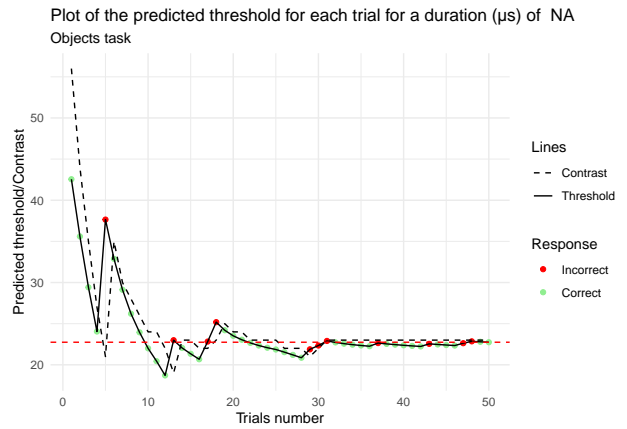
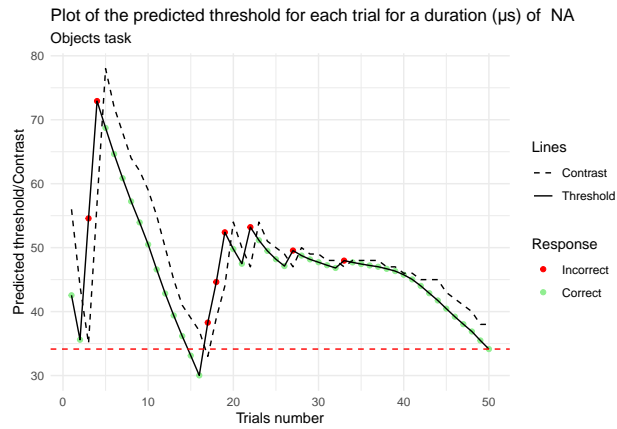
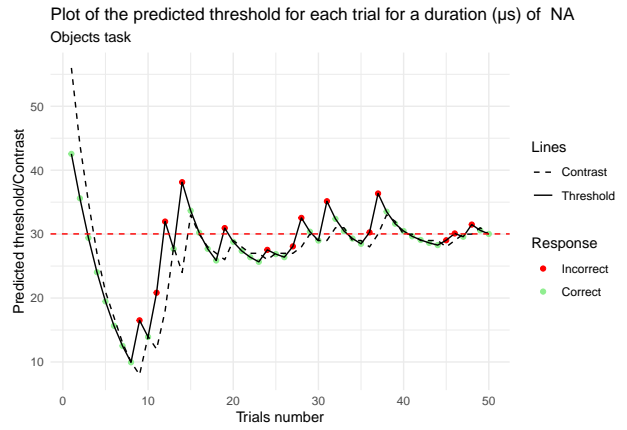
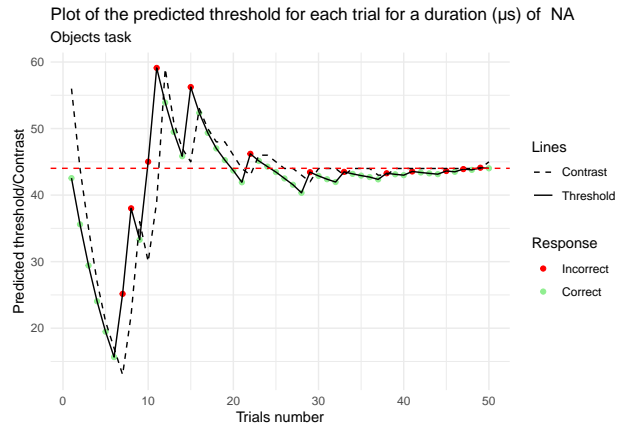
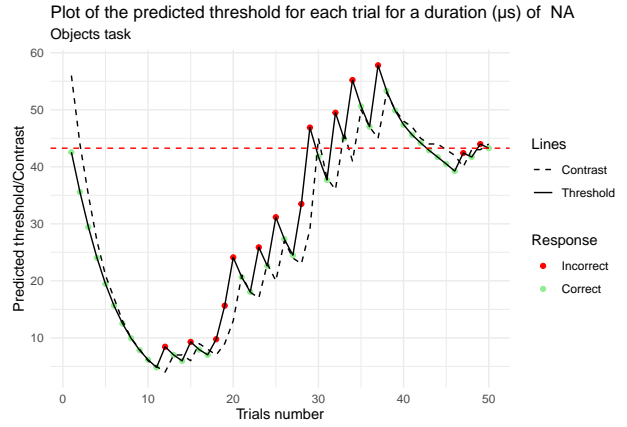
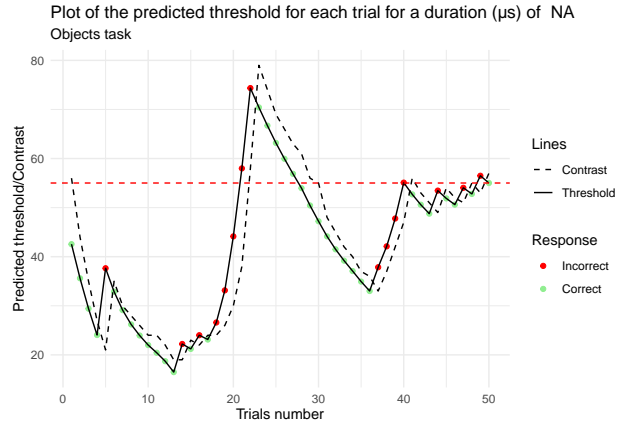
We divide the three data frames into subsets for eachs 10 durations and we extract the final threshold for each of theses durations and we also extract the final contrast for each of theses durations.

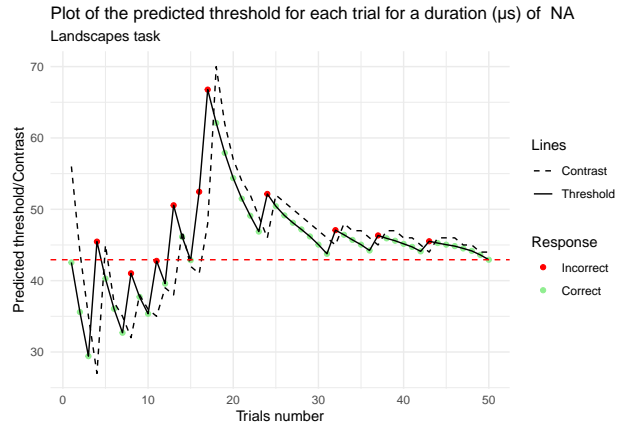
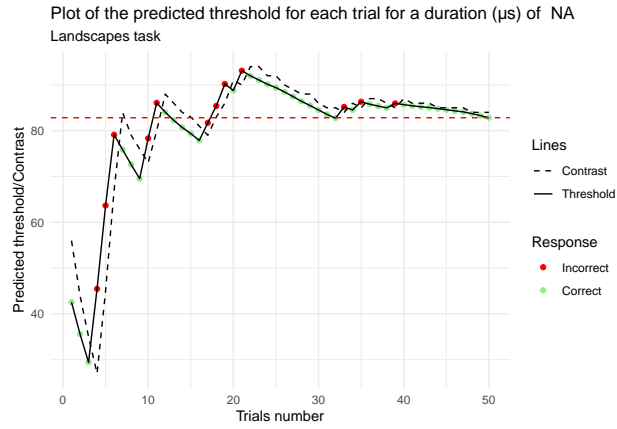
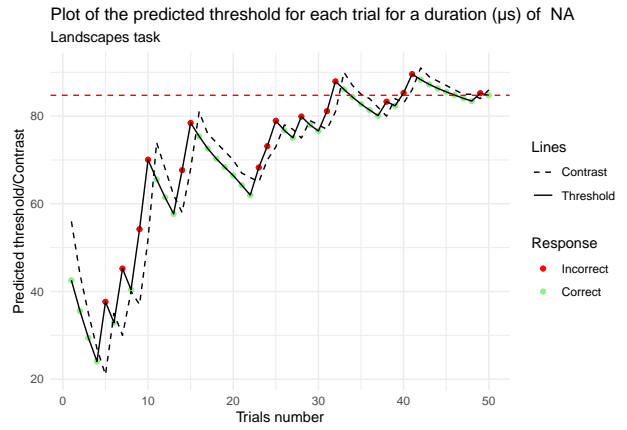
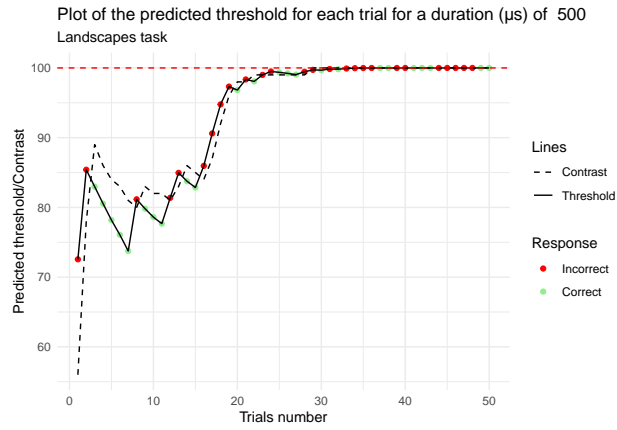
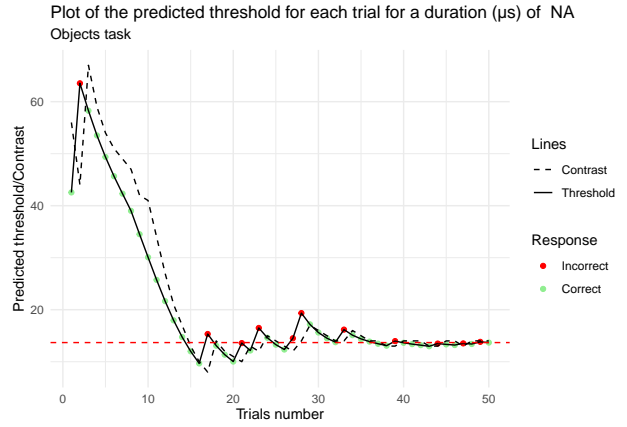
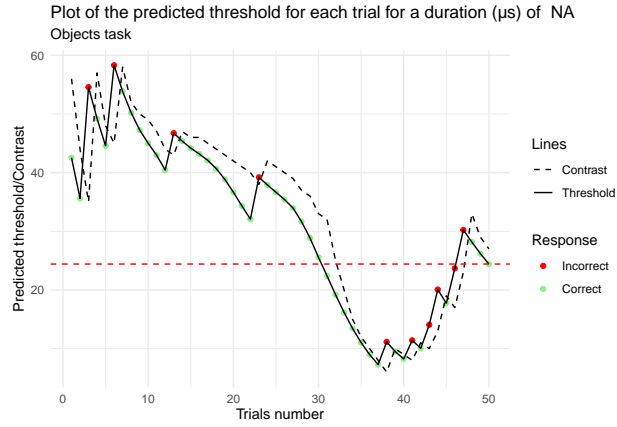
Stairs plots

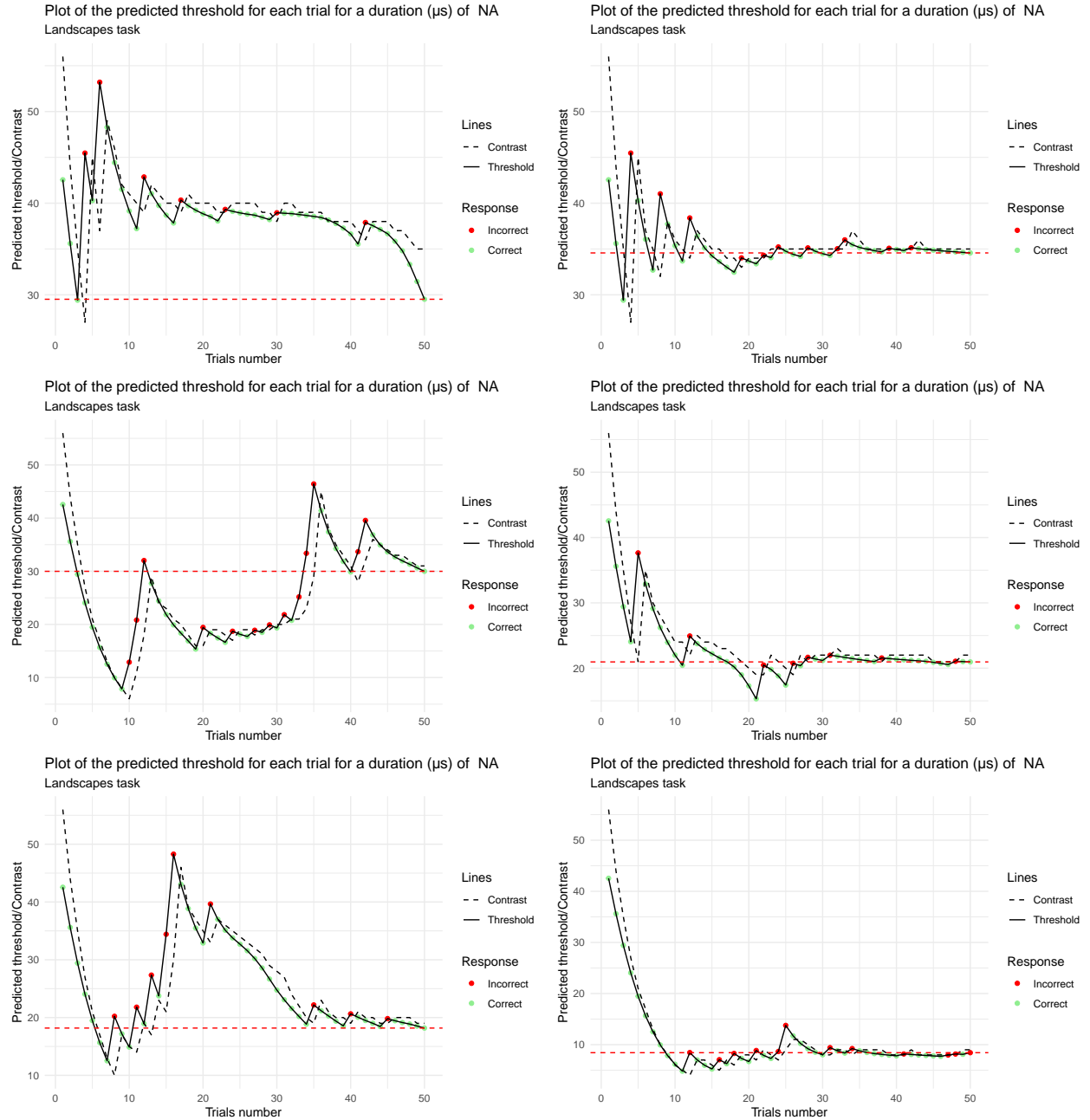
Now we create plot of updated threshold for each trials of each duration to see if the algorithm is working. If it worked we should see stair like plot that converge to a certain value at the 50th trials. If we dont see any stair shape it mean that it didn't work properly, and if the value to which the updated threshold converge is 100 or > 99 it mean that we can't find the "true" threshold within our contrasts range. So theses value will be filtered later.





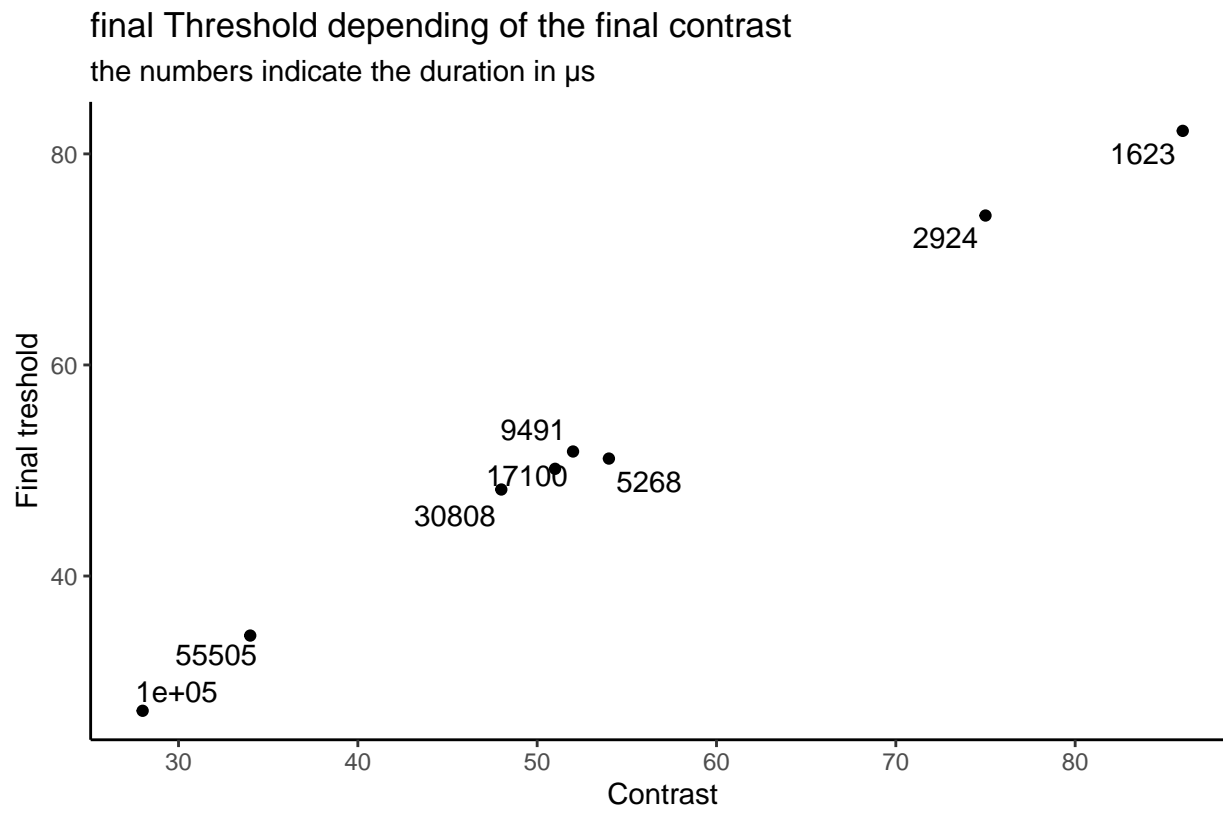






Filtering the threshold

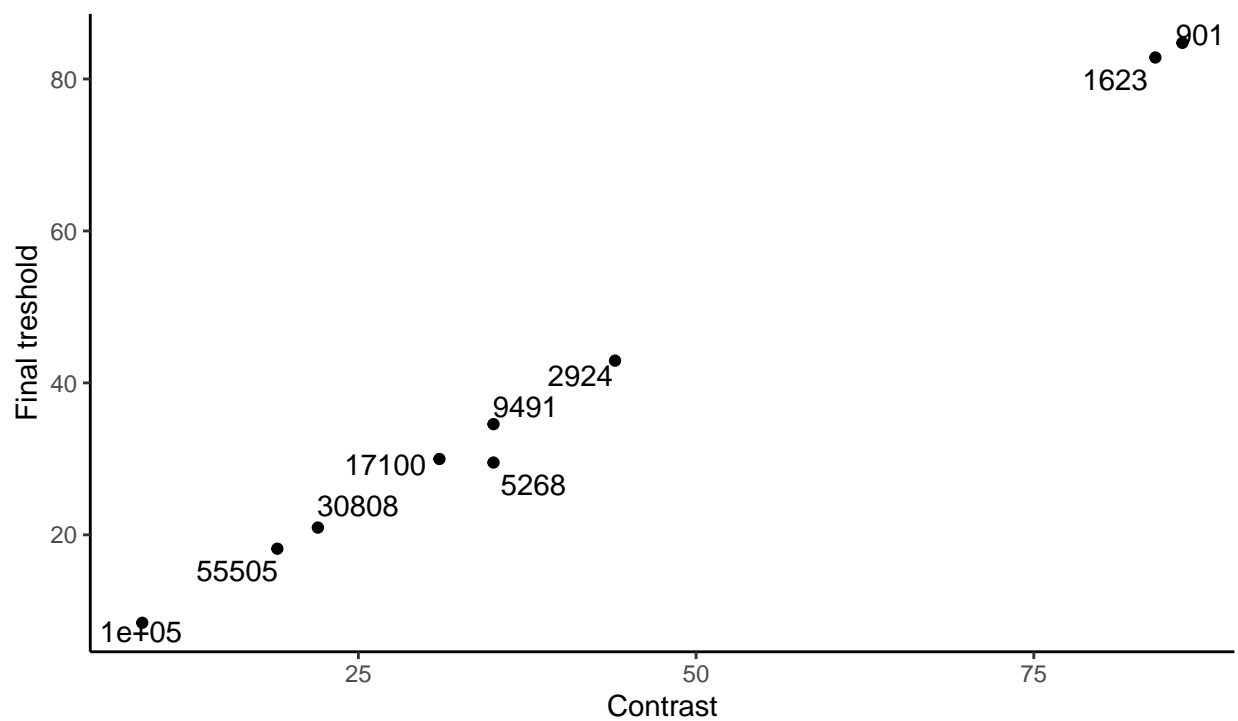
We are first creating a data frame which contain three columns : final_thresholds; finals_contrasts; Durations. For each tasks. And then we are getting rid of the threshold that are equal or above 99 for reasons explain above



in these data frame

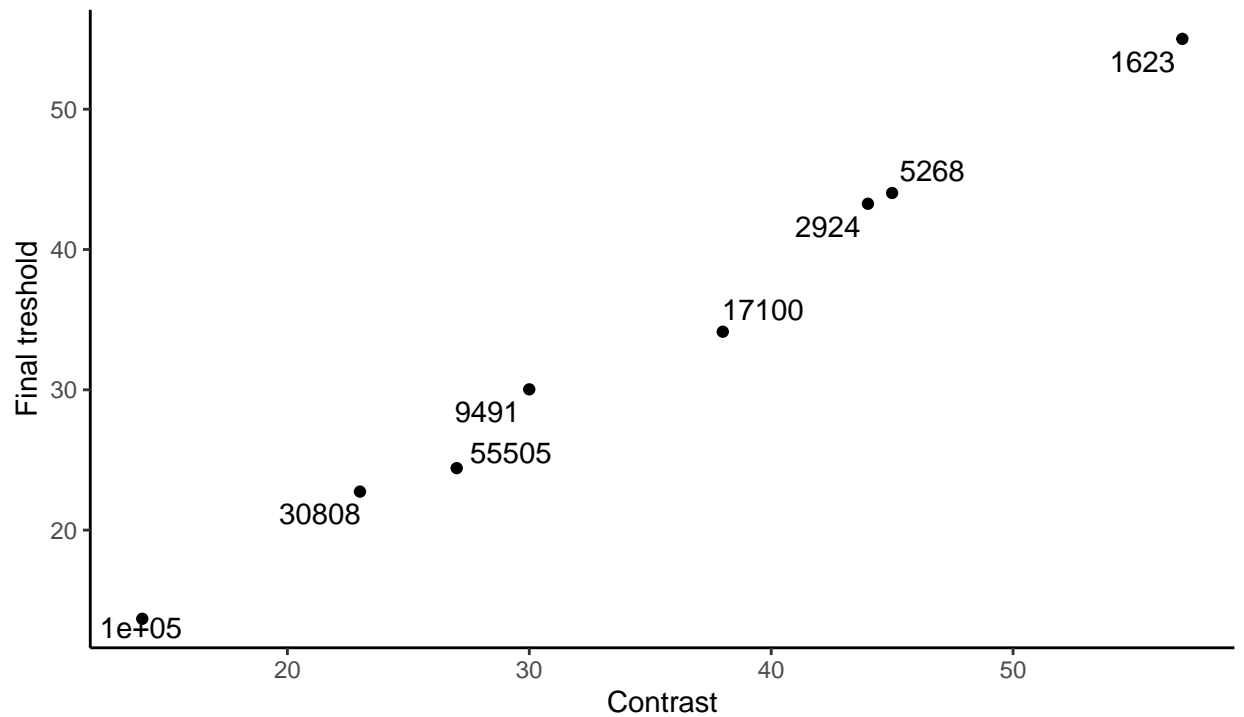
faces tas

final Threshold depending of the final contrast
the numbers indicate the duration in μs



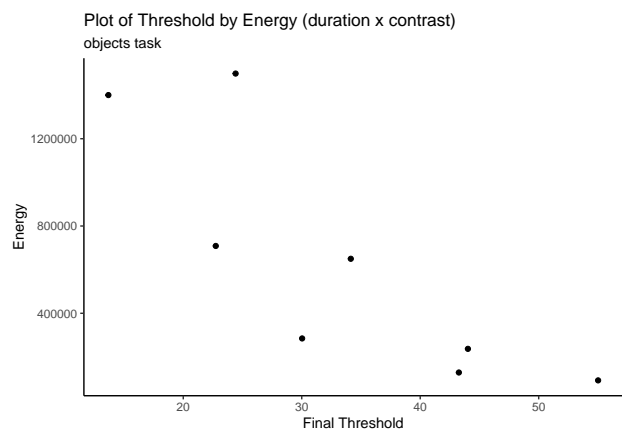
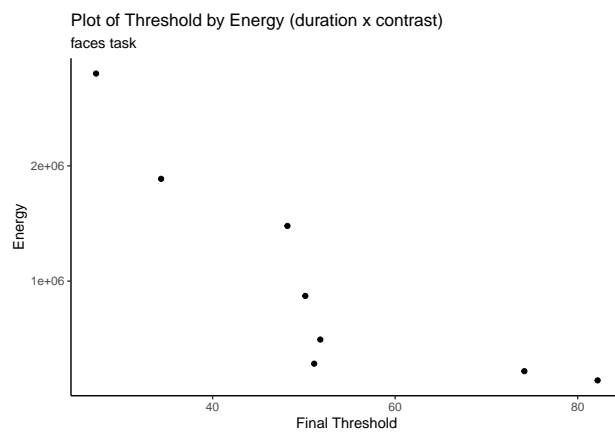
landscapes task

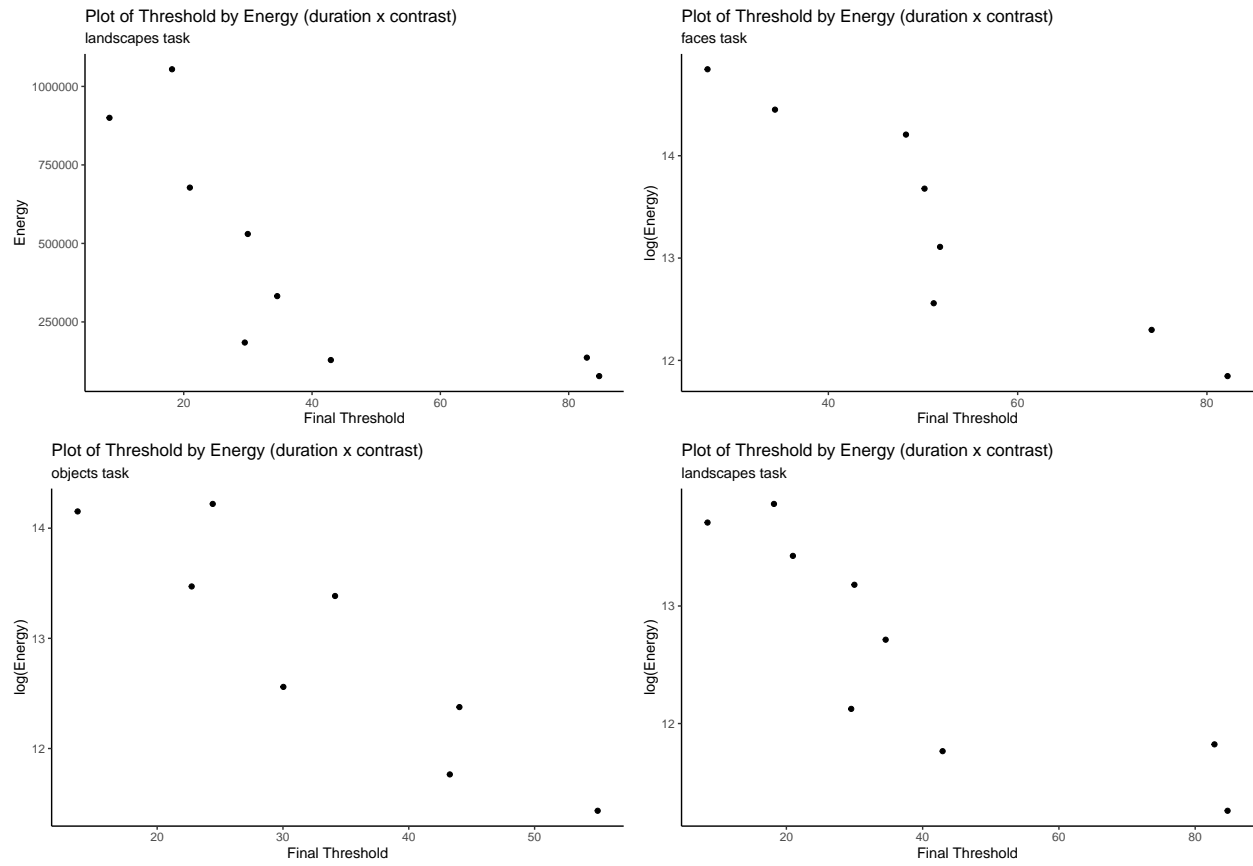
final Threshold depending of the final contrast
the numbers indicate the duration in μs



Ploting Energy

Now we can plot energy E and $\log(E)$ which is : $E = \text{duration} \times \text{contrast}$:





Energy by duration

Let's plot Energy by duration, which is easier to interpret.

